Cahiers GUTenberg

# BALISES, STRUCTURES ET TEI

Jacques André

*Cahiers GUTenberg*, nº 24 (1996), p. 11-22.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1996___24_11_0>

# Balises, structures et TEI

Jacques ANDRÉ

*Irisa/Inria-Rennes, campus de Beaulieu*
*F-35042 Rennes cedex*
jacques.andre@irisa.fr

**Résumé**. En introduction à la *TEI (Text Encoding Initiative)*, nous rappelons les principes de ce genre de codage (utilisation de balises parenthésées dans le texte permettant de marquer une ou plusieurs structurations de celui-ci). Nous insistons notamment sur la différence entre un tel codage « interne » et les vues « externes » qu'on peut en avoir (sur écran, lors de leur impression par un éditeur, etc.) et ceux proposés par des produits commerciaux comme Word. Ce balisage structuré n'a de sens que si des outils existent pour en aider la saisie, pour travailler sur ces structures et pour éditer, de façons diverses, ces documents.

## 1. Introduction

La *TEI (Text Encoding Initiative)* [1] est un système de codage permettant l'échange et la circulation de documents électroniques habituellement utilisés en sciences humaines. Ce codage utilise implicitement un certain nombre de concepts – balises, structures, SGML, parseurs, etc. – qui finissent par faire partie du « non-dit » des publications sur ce sujet et qui peuvent bloquer les chercheurs qui n'ont souvent été formés qu'à l'emploi d'outils de PAO, comme Word qui ne sont que des outils de PAO, c'est-à-dire des outils de mise en page et non d'échanges de documents.

### 1.1. Plusieurs visions d'une œuvre

Prenons un roman, par exemple *La vie mode d'emploi* de Georges Perec. Il peut être vu de nombreuses façons différentes.

1. Tout ce *Cahier GUTenberg* est consacré à ce sujet ; tout son contenu est/sera accessible à l'url : `http://www.univ-rennes1.fr/pub/GUTenberg/publications`
On y trouvera notamment une présentation de Nancy Ide et Jean Véronis [12] et, dans l'éditorial de François Role (pages 1–3), une liste des documents et URL à consulter. Ce *Cahier* contient par ailleurs une traduction française de la *TEI-Lite* qui est une version simplifiée de la TEI [4].

– pour le lecteur, c'est une œuvre qui existe sous forme de livre ou de cassette audio (pour les aveugles) et dont il existera bien un jour un film ;
– pour Gérard Genette [7], une telle œuvre ne comprend pas seulement le texte, mais aussi l'*épitexte* (interviews, etc.) et le *péritexte* (couverture, préface, etc.) ;
– un typographe ou un maquettiste en verra plutôt l'aspect physique : format de papier, format de page, polices, corps courant, interlignage, foliotage, etc. ; à partir du même « texte », Hachette et le Livre de poche n'ont pas produit la même chose imprimée ;
– un éditeur bilingue aura le besoin de respecter le sens de ce texte et le soucis de trouver une correspondance visuelle entre le texte original et le texte traduit ;
– un biliothécaire et un libraire verront des informations telles que le titre, l'auteur, l'éditeur, le format du livre, son ISSBN, son URL, son prix, etc.
– un généticien (travaillant sur la genèse des œuvres [10]) étudiera plutôt les carnets manuscrits de Perec [11] ;
– un oulipien verra surtout la structure de « bi-carré latin orthogonal d'ordre 10 » [11] sous-jacent à l'ordonnancement des chapitres et des personnages de ce roman ;
– d'autres chercheurs (bibliologues, géographes, botanistes, musicologues, etc.) pourraient avoir encore des visions différentes.

Ce qui ressort de cette énumération, c'est que la chose imprimée n'est que la partie visible des œuvres littéraires, mais non la seule digne d'échanges !

### 1.2. Principes de la TEI

– La TEI est un codage qui, sans être universel, se veut quand-même assez général pour intéresser le plus grand nombre possible de classes de chercheurs, éditeurs, etc. Disons dès à présent que certains points ne sont pas, ou peu, abordés aujourd'hui par la TEI (où T signifie *Text* !), par exemple tout ce qui touche au contenu des images et au son.

– Ce codage se veut indépendant des « plateformes », c'est-à-dire indépendant de telle ou telle marque d'ordinateur, de tel ou tel système commercial de manipulation de texte, mais aussi de tel ou tel réseau (car il s'agit bien d'échanger des documents entre chercheurs éloignés). Un corollaire de ceci est malheureusement un certain nivellement par la base (par exemple l'emploi du code Ascii).

– Ce codage est un codage interne[2]. Ce n'est pas obligatoirement la forme sous laquelle les chercheurs doivent l'utiliser. En particulier, divers types d'outils (ou diverses fonctionnalités d'un même produit) sont à employer pour travailler confortablement avec la TEI (en ignorant les balises) :

2. Sous-entendu à l'informatique ; l'erreur habituelle est probablement de présenter cette TEI en termes de codage interne et non avec des mots compréhensibles par l'utilisateur ; d'où cette note ...

– un système de saisie,
– un système d'affichage sur écran,
– un système d'édition,
– un système de parcours dans le texte, etc.

# 2. Marquage et balises

Depuis longtemps, on a l'habitude de mettre dans un texte (au sens de la surface écrite) des marques, connues sous divers noms tels que « notes tironiennes », « gloses », « notes marginales », « annotations », etc. Le marquage électronique relève du même esprit : il s'agit d'insérer, non plus dans la « surface » d'une page de *codex*, mais dans un fichier électronique (que l'on peut considérer, a priori, comme linéaire [3]) des informations liées au texte lui-même mais n'en faisant pas directement partie. Comme il s'est trouvé que les premières bandes perforées textuelles ont été utilisées pour commander des Linotypes puis des photocomposeuses, les premiers marquages électroniques ont concerné des commandes typographiques (passer en gras, changer de corps, faire un retrait, etc.). Mais, les textes ayant été très tôt l'objet de recherches en informatique [4], il n'est pas surprenant que le marquage des textes ait pris aussi une connotation linguistique au sens large.

## 2.1. Diverses formes de balises

Ce marquage se fait à l'aide de «balises» (en anglais *tags* ou *marks*). La méthode la plus fréquente est d'insérer comme balise un texte entre des marqueurs spécifiques ; exemples :

– photocomposeuse GS400 : `On change de police [CF1] ici ...`
– TeX : `\noindent Ce paragraphe n'a pas de retrait ...`
– RTF (*Rich text Format*, codage interne à Word) : `Si je {\ul souligne ...`
– TEI : `Dans <title>La jeune Parque ...`

ce qui implique bien sûr que si, selon le cas, l'un de ces caractères `\{<>[]` est employé dans le texte, il doit alors être codé.

Beaucoup de balises indiquent des changements d'état (par exemple passer en italique) et il est naturel d'offrir le retour à l'état inverse (par exemple re-

3. Mais si les bandes perforées puis magnétiques étaient effectivement linéaires, ce n'est bien sûr plus du tout le cas des textes en mémoire !

4. Il suffit de citer Vannevar Bush, « inventeur » des hypertextes dès 1945 [3] et de rappeler que des langages de manipulation de chaînes existaient dès 1956 (Fortran ne date que de 1955). En France, dès 1965, Nancy était le siège de deux équipes de linguistique informatique : celle du *Trésor de la langue française* et le Graal où, entre autres, la *Bible* a été saisie sur cartes perforées avec des annotations vocaliques, syntaxiques, voire talmudiques.

prendre la fonte précédent l'italique). Les balises sont donc parenthésées : soit il existe deux balises, l'une pour ouvrir le nouvel état et l'autre pour le fermer, exemples :

– TEI : Dans <title>La jeune Parque<\title>, on lit ...

soit un symbole indique la fin de la portée de la balise, par exemple :

– RTF : Si je {\ul souligne} un mot ...

## 2.2. Éditeurs et balises

Il existe deux sortes d'éditeurs : ceux qui nécessitent une manipulation explicite et ceux qui utilisent un codage implicite en balises, mais sans que l'utilisateur le voit. (La)TeX, par exemple, relève du premier cas car il demande qu'on lui fournisse un texte balisé. La saisie se fait alors avec un éditeur classique, style emacs. En revanche, Word permet à l'utilisateur de travailler sans voir les balises de RTF qui existent pourtant bien de façon « interne »[5]. Word peut en effet être considéré comme formé de plusieurs programmes coopérants, cet ensemble de programmes donnant à l'utilisateur l'illusion de travailler directement sur la forme finale de son document[6] :

1. un programme de saisie/correction qui enregistre, dans un fichier « interne », le texte tapé et qui traite les commandes du type « cliquer » (par exemple, dans le menu « format », sur l'entrée « souligné ») en insérant dans ce texte interne les balises correspondantes ({\ul et }) ;

2. un programme d'affichage sur écran d'un texte balisé dans le code RTF ;

3. un programme d'impression qui lit ce fichier RTF, le transforme en fichier PostScript et l'envoie à une imprimante ;

4. un programme qui parcourt le fichier RTF en ignorant les balises[7].

Si nous nous sommes un peu étendu sur cet aspect, c'est pour faire comprendre que

5. Mais l'utilisateur peut y accéder : il suffit, par exemple avec un Macintosh, de « cliquer » sur « enregistrer sous » puis dans « format » puis dans « RTF » pour voir le texte balisé.

6. D'où l'expression, un peu désuète, de WYSIWYG, *What You See Is What You Get*, que l'on a longtemps opposée, à tort (voir figure 1), aux éditeurs structurés !

7. Il est appelé, par exemple, lorsque l'on fait une « recherche » : en effet si on cherchait les deux lettre « ul » dans ce fichier et s'il y avait du texte souligné, on trouverait « ul » dans la balise \ul ; de même, si on cherche un mot qui est coupé en fin de ligne on le trouvera bien. Par contre RTF ne permet pas de marquer simplement les ligatures : si on cherche la séquence « fin » et si on a utilisé la ligature « fi » dans le mot « définition », celui-ci ne rendra pas la réponse positive espérée.

– la TEI n'est pas un éditeur ou un formateur comme Word ;
– la TEI est une norme de codage de textes balisés, comme RTF ;
– pour utiliser confortablement un texte codé en TEI, sans avoir à manipuler de balises, il est préférable d'utiliser un outil spécialisé ;
– tout texte codé en TEI est complètement indépendant des outils utilisés ;
– la TEI est, de plus, un codage de documents « structuré ».

## 2.3. Création de nouvelles balises – types

Les formateurs permettent, en général, de créer de nouvelles balises. En LaTeX, par exemple, la commande

```
\newcommand{\latin}[1]{\textit{#1}}
```

définit une balise telle que si l'on écrit `\latin{Sator arepo...}` cette expression sera mise en italique. De même, Word offre la possibilité de définir de nouvelles commandes. Toutefois, ceci se situe à un niveau en amont de RTF : la commande équivalente à celle `\latin` ci-dessus ne sera pas incluse dans RTF (on n'y retrouve notamment pas le nom `latin`), mais traduite de suite en balises RTF de base : `{\it ...}` .

En fait ces déclaration de nouvelles balises parenthésées correspondent à la définition de nouveaux éléments, c'est-à-dire de « types » en terme de langage de programmation. Ces types sont construits à partir des types élémentaires, prédéfinis, et peuvent profiter des propriétés d'héritage (par exemple Word permet de définir un nouveau style `titre` à partir de : « `standard +` » ; un `soustitre` peut à son tour être défini à partir de `titre`, etc.). En Word, cette définition de type s'appelle «style» car ses types élémentaires sont uniquement graphiques, c'est-à-dire liés à l'aspect visuel du document, à son style.

## 2.4. Références et liens hypertextuels

Word permet de faire des notes en bas de page. Mais si l'on regarde le texte RTF correspondant, on voit qu'il sagit simplement d'une insertion de la note là où elle est appelée. Il n'est pas facilement possible d'appeler la même note automatiquement (c'est-à-dire sans recopier le numéro) depuis plusieurs endroits. À plus forte raison, est-il difficile d'utiliser un double système de notes dont l'un pour l'apparat critique (voir [16] dans ce *Cahier*).
En revanche, de nombreux systèmes de balisage, comme LaTeX ou la TEI, permettent des renvois d'un point (grace par exemple à un balisage du genre `<pointer>sur_quoi</pointer>` (où `sur_quoi` est un endroit du texte qui a été marqué par exemple par un balisage `<ancre>sur_quoi</ancre>`). Ces balisages permettent, de plus, de préciser si l'on veut indiquer le numéro

de la page ou de la section, ou autre chose. Enfin, on peut en général « typer » ces pointeurs.

Cette notion de pointeur est à base des systèmes hypertextes.

# 3. Documents structurés

Dans les années 1970–1985, des utilisateurs (des éditeurs – en France le Syndicat National de l'Edition et, aux USA, l'*American Association of Publishers*, des spécialistes de documentation technique, etc.) ont essayé de définir des normes de balisage ; ça a été laborieux, mais quelques produits ont émergé, comme GML *(General Markup Language)* d'IBM, la famille `troff, nroff`, etc. de la Bell et TEX de Knuth. Mais aucun de ces formateurs n'était un formateur structuré. C'est probablement Brian Reid qui a défini le premier langage de document structuré, *Scribe*. Les principes des documents structurés ont été abondamment décrits dans [1, 2, 5]. En voici les principaux.

## 3.1. Séparation logique/physique

Le principe de base est «à chacun son métier ...» : aux auteurs de travailler sur le «contenu» de leurs œuvres et aux typographes ou maquettistes de régler les problèmes de forme. Ceci se fait par deux jeux distincts de balises : logiques (chapitres, sections, dialogues, textes, etc.) et physiques (mise en page, polices, etc.), ces dernières s'appuyant sur les premières. À partir d'une même œuvre logique, *La vie mode d'emploi* par exemple, on imprime (par simple changement du rapport logique/physique) deux livres différents, respectant la maquette de la collection « Hachette littérature » ou bien celle du « Livre de poche ».
Réciproquement, à une mise en page donnée peuvent correspondre des structures logiques aussi différentes que (comme dans l'encadré ci-après) un « nom de journal » *(Le Monde)* ou une définition *(fonte)*, un dialogue ou une liste, etc. ce que Word ne saurait faire qu'à partir du seul aspect visuel !

| | |
|---|---|
| Jean lisait *Le Monde* quand brusquement entra Nicole :<br>– on va être en retard si tu continues à lire, dit elle,<br>– je suis prêt dans 5 mn, cria-t-il,<br>mais en fait, il finit la lecture de son article bien avant. Ils ... | On appelle *fonte* électronique l'ensemble formé :<br>– des algorithmes de dessin des caractères,<br>– des tables de métrique associées,<br>comme du temps de la composition au plomb. On peut ... |

## 3.2. Généricité

Une définition structurée de documents permet de décrire l'organisation logique non pas d'un seul document *spécifique*, par exemple le Que-sais je ? n° 685 de Nina Catach sur *L'orthographe*, mais de tous ceux de la classe *générique* des Que-sais je ?

La TEI va même plus loin car elle propose non seulement de s'adresser à une classe d'ouvrages, mais même à des choses aussi variées que des pièces de théâtre, des œuvres poétiques, des textes juridiques, etc. La difficulté est bien sûr de trouver un système qui soit à la fois minimal (c'est-à-dire d'avoir le minimum de définitions de base sans être un « pneu couvert de rustines ») et complet (c'est-à-dire de traiter le maximum de choses). En ce qui concerne la TEI, cette approche a été faite non pas par des informaticiens ni même par des «éditeurs», mais par des philologues qui ont ainsi construit un inventaire – une sorte de « flore », au sens de Buffon – des divers éléments pouvant constituer un document littéraire [8].

## 3.3. Meta-langages

En fait, la grande nouveauté des documents structurés est surtout d'utiliser un langage pour définir l'usage des balises. Les linguistes, notamment depuis Chomsky, savent bien décrire des règles grammaticales comme :

```
      <Phrase> = <GN> <GV>
      <GN> = <Art> <Adj>* <Nom>
      <GV> = <Verbe><Adv> | ...
      <Art> =  le | la | un | ...
```

Avec cette grammaire et en parenthésant les éléments internes, « Le joli chien court rapidement » s'écrit :

```
<Phrase>
   <GN><Art>Le</Art><Adj>joli</Adj><Nom>chien</Nom></GN>
   <GV><Verbe>court</Verbe><Adv>rapidement</Adv></GV>
</Phrase>
```

8. Même si les livres ont fait l'objet d'études nombreuses tant sur le plan typographique et les structures sémantiques associées [1, pages 143–159] que sur celui bibliologique (Laufer), cette « flore » restait à faire et c'est au moins un apport très positif de la TEI d'avoir, dans ses *base tag sets*, établi cette liste d'éléments de base.

Une telle notation a été naturellement employée à un niveau plus large que celui de la phrase et on retrouve donc dans toutes les définitions de documents structurés des notations comme la suivante :

```
<roman> = <chapitre>*
<chapitre> = <titre><paragraphe>*
<paragraphe> = <texte>
<titre> = <texte>
```

qui dit donc, simplement, qu'un roman est formé de chapitres, qu'un chapitre est formé d'un titre et d'un nombre quelconque de paragraphes et qu'un paragraphe c'est du texte [9]. La seule ressemblance avec les documents balisés comme RTF/Word n'est, finalement, que l'emploi de balises (celles de SGML/TEI marquant des structures logiques « profondes », celles de Word des structures visuelles «de surface»).

# 4. Langages de documents structurés

Divers langages de documents structurés ont été ainsi définis (on trouvera dans [5] une bibliographie à jour sur ce sujet) dont LaTeX [10], Grif [11] et SGML.

## 4.1. SGML

SGML est un langage de balisage basé sur GML qui a été réécrit pour satisfaire les principes des documents structurés (voir [8, 9, 17]). SGML se trouve actuellement très répandu du fait que l'armée américaine l'a imposé comme norme pour la documentation des matériels qui lui sont livrés (CALS) et parce que

9. Toutefois, entrer à un niveau aussi bas que la structure grammaticale de la phrase, même si indispensable pour certaines applications, pose des problèmes de saisie – à chaque balise du type `<GN>` il faudrait «taper» le nom de la balise et de celle la fermant – aussi les documents structurés (mais ce n'est pas une obligation) s'arrêtent en général au niveau du paragraphe («texte») et non à celui plus interne de la phrases et de ses structures grammaticales.

10. LaTeX est un peu spécial dans la mesure où, si la séparation physique/logique est bien nette, la notion de meta-langage permettant de définir des classes de documents est très mélangée avec celle de structure physique, à tel point que très peu de personnes sont vraiment capables de définir de nouvelles classes de documents autrement que par héritage de classes prédéfinies.

11. Grif était le nom d'un prototype défini à l'Inria [6] dont il existe une version commerciale ; il s'appelle maintenant Thot [13, 15] et il lui correspond un *browser* domaine public, Amaya de W3 : `http://opera.inrialpes.fr/OPERA` ; la figure 1 a été faite avec cet éditeur structuré.

le réseau Internet utilise depuis peu une « DTD » simplifiée, en l'occurrence HTML.

SGML est en fait un meta-langage permettant de décrire des grammaires définissant des classes de documents structurés. Il y a donc plusieurs niveaux (voir [17] pour plus de détails techniques) :

1. La grammaire `<Phrase>= <GN>...` donnée ci-dessus en 3.3 utilise implicitement un certain nombre de constructions : `<GN><GV>` par exemple signifie qu'un élément `<GN>` doit être suivi d'un élément `<GV>` ; `<Adj>*` signifie qu'un élément peut ne pas apparaître, apparaître une fois, ou un nombre illimité de fois ; `|` signifie «ou» ; enfin, le vocabulaire terminal `le | la | un | ...` est décrit d'une certaine façon. SGML propose donc un premier niveau où de telles constructions sont proposées.

2. La grammaire définissant une classe générique de documents (par exemple un roman) peut être décrite en utilisant ces constructions. Cette grammaire s'appelle une DTD *(Document Type Definition)* et définit donc ainsi les balises d'une classe de documents. On pourrait écrire :

```
     <!ELEMENT roman (chapitre+) >
     <!ELEMENT chapitre (titre, paragraphe+)>
     <!ELEMENT paragraphe (texte) >
     <!ELEMENT titre ( texte) >
```

3. Chaque document spécifique, ici un roman de Perec, doit alors être codé avec des balises qui respectent la DTD considérée, par exemple :

```
<roman><chapitre><titre><texte>Dans
                          l'escalier</texte></titre>
<paragraphe><texte>Oui, cela pourrait commencer...
... combles.</texte></paragraphe>...
... </texte></paragraphe></chapitre></roman>
```

## 4.2. DTD de SGML

De nombreuses DTD ont été écrites pour des classes de documents précises (par exemple une DTD pour le rapport annuel de l'Inria). Mais, concevoir une classe de documents étant un travail très minutieux, surtout si l'on veut garder le plus de généralité, ces DTD sont souvent définies par des organismes privés ou publics (tels que les syndicats d'éditeurs, le service des publications de la

CEE, l'office international des brevets, etc.). C'est le cas de la TEI (voir [12]). Une DTD, toutefois, est un peu spéciale [12] : HTML.

### 4.3. Attributs

Les éléments manipulés par SGML et donc la TEI, sont essentiellement liés à la structure du texte. On a souvent besoin d'y associer aussi une certaine sémantique, ne serait-ce qu'en distinguant des noms de balise, par exemple en écrivant :

```
<nom-propre>= <nom-de-lieu>| <nom-de-personne>| <nom-d-oeuvre>
<nom-de-lieu>= <nom>
nom-d-oeuvre>= <nom>
<nom-de-personne>>= <nom>
<nom>= <texte>...
```

Ceci se fait de façon condensée en «attribuant» les balises, par exemple en ne définisant que <nom propre> mais en lui passant un paramètre au moment de l'appel. On pourra alors coder du texte comme :

```
 Le roman <nom-propre type=oeuvre>La vie mode
d'emploi</nom-propre>a été
 écrit par <nom-propre type=personne>Georges Perec</nom-propre>
 à <nom-propre type=lieu>Paris</nom-propre>de 1973 à 1978.
```

Cette notion, utilisée plus richement dans la TEI, permet de réduire considérablement la taille de cette DTD.

### 4.4. Parseurs

Une des principales forces des documents structurés est d'empècher un certain nombre d'erreurs, par exemple de mettre un titre au milieu d'un chapitre (ce que Word ne peut empécher de faire !). Un texte balisé comme suit

```
<paragraphe> ... <titre> ... </titre> ... </paragraphe>
```

devra donc être interdit. La seule façon de vérifier ceci est évidemment d'employer un analyseur syntaxique comme ceux qui vérifient que «Le joli mange chien» n'est pas syntaxiquement correct. Ces analyseurs s'appellent des «parseurs» *(parsers)* car ils s'appuient sur les balises (et donc sur une DTD, la TEI par exemple) mais sans «analyser» le contenu du texte lui-même. [9] cite plusieurs de ces analyseurs.

12. Par son très petit nombre de règles (elle est donnée en annexe de [9]) et par le fait que ses éléments logiques étant de très bas niveau, les utilisateurs s'en servent plus comme une norme de document physique que logique, contrairement à l'esprit même de HTML !

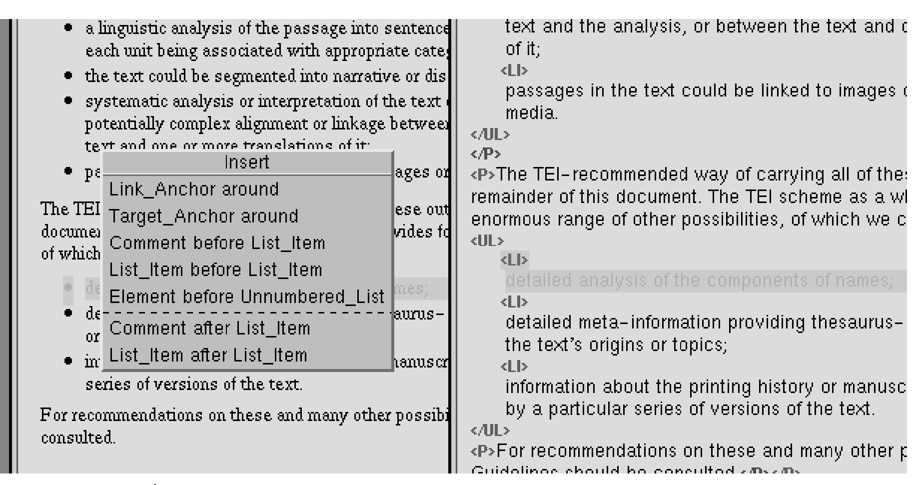



FIGURE 1 – *Écran d'un éditeur SGML : à gauche, une vue logique (le système, par exemple, ne propose pas d'insérer un « chapitre » à cet endroit, puisque la DTD ne l'autorise pas), à droite le même document « interne » avec ses balises.*

## 5. Conclusion

Les documents structurés, et la TEI en particulier, permettent de prendre du recul vis-à-vis de la forme extérieure, de surface, des textes et vis-à-vis des moyens (réseaux, plateformes, etc.) d'échange. Nous avons volontairement caché les possibilités « graphiques » liées aux documents structurés, comme SPDL, les *Cascading Styles Sheets*, etc. nous contentant ici de renvoyer à [8, 14, 15].

Nous espérons simplement ici avoir permis à certains lecteurs de mieux aborder la lecture de ce *Cahier* sur la TEI et la documentation complète de cette DTD qui se veut très ouverte aux chercheurs en sciences humaines.

## Bibliographie

[1] Jacques ANDRÉ, Richard FURUTA, and Vincent QUINT, *Structured documents*, Cambridge University Press, 1987.

[2] Jacques ANDRÉ et Vincent QUINT, « Structures et modèles de documents », in *Le document électronique* (Christian BORNE ed.), Inria, 1990, 3–57.

[3] Colin BURKE, *Vannevar Bush, ULTRA and the other MEMEX*, Scarecrow Press, Londres, 1994.

[4] Lou BURNARD et M.C. SPERBERG-MCQUEEN (traduction française de François ROLE), « La TEI simplifiée : une introduction au codage des textes électroniques en vue de leur échange », *Cahiers GUTenberg* n° 24 (ce numéro), juin 1996, 23–151.

[5] Robin COVER (ed.), *A bibliography on structured texts*, Technical Report n° 281, Queens University, Kingston, Canada, 1990.
`http://www.sil.org/sgml/sgml.html`

[6] Richard FURUTA, Vincent QUINT, and Jacques ANDRÉ, « Interactively Editing Structured Documents », *Epodd – Electronic Publishing*, vol. 1, n° 1, avril 1988.

[7] Gérard GENETTE, *Seuils*, Le Seuil, Paris, 1987.

[8] Michel GOOSSENS et Éric VAN HERWIJNEN, « Introduction à SGML, DSSL et SPDL », *Cahiers GUTenberg*, n° 12, décembre 1991, 37–56.

[9] Michel GOOSSENS, « Introduction pratique à SGML », *Cahiers GUTenberg*, n° 19, janvier 1995, 27–58.

[10] Almuth GRÉSILLON, *Éléments de critique génétique – lire les manuscrits modernes*, PUF, Paris 1994.

[11] Hans HARTJE, Bernard MAGNÉ et Jacques NEEFS (présentation, transcription et notes de), *Cahier des charges de* La Vie mode d'emploi *de Georges Perec*, CNRS Editions et Zulma, Paris 1993.

[12] Nancy IDE et Jean VÉRONIS, « Présentation de la *TEI : Text Encoding Inititiative* », *Cahiers GUTenberg* n° 24 (ce numéro), juin 1996, 4–10.

[13] Vincent QUINT, Hélène RICHY, Cécile ROISIN et Irène VATTON, « Thot, manuel utilisateur », Inria, Janvier 1966.

[14] Hélène RICHY, Chrystèle HÉRAULT et Jacques ANDRÉ, « Notion de feuille de dstyle », *Cahiers GUTenberg*, n° 21, juin 1995, 127–134.

[15] Cécile ROISIN and Irène VATTON, « Merging Logical and Physical Structures in Documents », *Epodd – Electronic Publishing*, vol. 6, n° 4, avril 1994, 327–337.

[16] François ROLE, « Le codage informatique des apparats critiques : évaluation des recommandations de la *Text Encoding Initiative* », *Cahiers GUTenberg* n° 24 (ce numéro), 154–165.

[17] Eric VAN HERVIJNEN, *SGML pratique*, International Thomson Pubishing France, 1995 .